

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-024837

(43)Date of publication of application : 25.01.2002

(51)Int.Cl.

G06T 7/40

G06T 7/00

H04N 1/40

(21)Application number : 2000-212362

(71)Applicant : FUJI XEROX CO LTD

(22)Date of filing : 13.07.2000

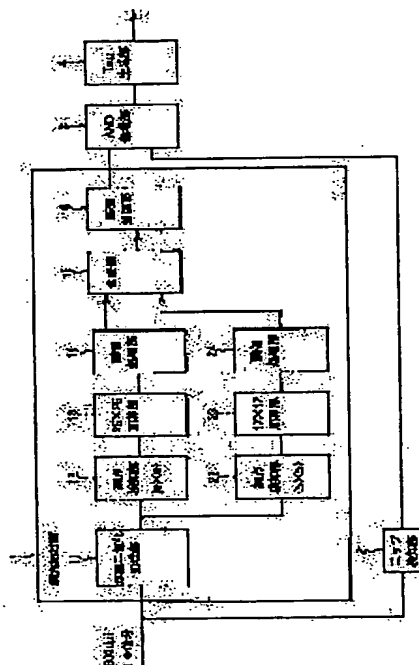
(72)Inventor : FURUYAMA KENJI

(54) IMAGE PROCESSING DEVICE AND METHOD, AND COMPUTER- READABLE RECORDING MEDIUM WITH IMAGE PROCESSING PROGRAM RECORDED ON THE MEDIUM

(57)Abstract:

PROBLEM TO BE SOLVED: To detect a dot region, even if its screen angle and screen lines are different from each other.

SOLUTION: This image processing device for detecting a dot region from an input image is provided with a plurality of dot detecting sections 12 and 22 for calculating a value, corresponding to the likeness of a dot in the specified region of the input image by window size different from each other or screen lines which are different from each other and threshold processing sections 14 and 24 for judging whether a prescribed region is a dot region, based on the value based on the likeness of the dots calculated by the plural dots 11 and 22 respectively.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開 2002-24837

(P 2002-24837 A)

(43) 公開日 平成14年1月25日 (2002. 1. 25)

(51) Int. Cl. 7	識別記号	F I	テーマコード (参考)
G 0 6 T	7/40	1 0 0	C 5C077
	7/00	2 0 0	C 5L096
H 0 4 N	1/40		F

審査請求 未請求 請求項の数 9

OL

(全 19 頁)

(21) 出願番号 特願2000-212362 (P2000-212362)

(22) 出願日 平成12年7月13日 (2000. 7. 13)

(71) 出願人 000005496

富士ゼロックス株式会社

東京都港区赤坂二丁目17番22号

(72) 発明者 古山 健司

神奈川県海老名市本郷2274番地 富士ゼロ

ックス株式会社海老名事業所内

(74) 代理人 100086298

弁理士 船橋 國則

F ターム (参考) 5C077 MP02 MP06 MP08 PP27 PP36

PP43 PP49 PP68 PQ12 PQ17

PQ20 RR02 RR15 SS01

5L096 AA02 AA06 BA08 FA06 FA43

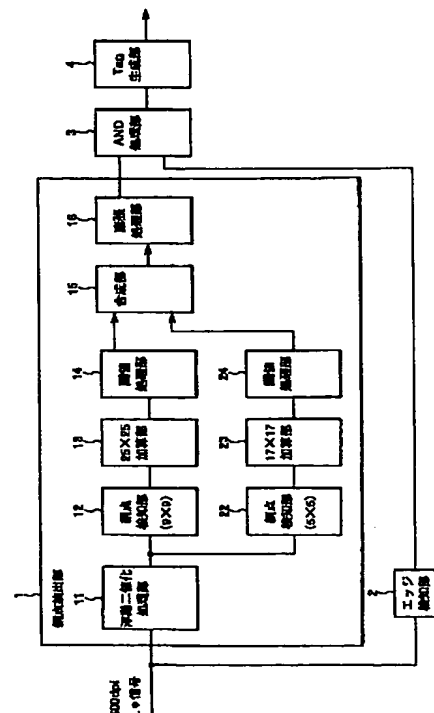
GA17 GA23 GA24 GA28 GA51

(54) 【発明の名称】 画像処理装置、画像処理方法および画像処理プログラムを記録したコンピュータ読み取り可能な記録媒体

(57) 【要約】

【課題】 網点のスクリーン角度やスクリーン線数が異なる場合でも的確に網点領域であることを検出できるようにすること。

【解決手段】 本発明は、入力画像から網点領域を検出する画像処理装置であり、各々異なるウィンドウサイズもしくは各々異なるスクリーン線数によって入力画像の所定領域の網点らしさに対応する値を算出する複数の網点検知部 12、22と、複数の網点検知部 11、22で各々算出される網点らしさに対応する値に基づき所定領域が網点領域であるか否かを判断する閾値処理部 14、24とを備えている。



【特許請求の範囲】

【請求項 1】 入力画像から網点領域を検出する画像処理装置において、

各々異なるウィンドウサイズによって前記入力画像の所定領域の網点らしさに対応する値を算出する複数の網点検知手段と、

前記複数の網点検知手段で各々算出される網点らしさに対応する値に基づき前記所定領域が網点領域であるか否かを判断する判断手段とを備えることを特徴とする画像処理装置。

【請求項 2】 入力画像から網点領域を検出する画像処理装置において、

各々異なるスクリーン線数によって前記入力画像の所定領域の網点らしさに対応する値を算出する複数の網点検知手段と、

前記複数の網点検知手段で各々算出される網点らしさに対応する値に基づき前記所定領域が網点領域であるか否かを判断する判断手段とを備えることを特徴とする画像処理装置。

【請求項 3】 前記複数の網点検知手段は、前記入力画像における孤立点を検知するにあたり、孤立点の中央付近ほど前記網点らしさに対応する値を高く設定することを特徴とする請求項 1 または 2 記載の画像処理装置。

【請求項 4】 入力画像から網点領域を検出する画像処理方法において、

各々異なるウィンドウサイズによって前記入力画像の所定領域の網点らしさに対応する値を算出する工程と、各々算出された網点らしさに対応する値に基づき前記所定領域が網点領域であるか否かを判断する工程とを備えることを特徴とする画像処理装置。

【請求項 5】 入力画像から網点領域を検出する画像処理方法において、

各々異なるスクリーン線数によって前記入力画像の所定領域の網点らしさに対応する値を算出する工程と、各々算出された網点らしさに対応する値に基づき前記所定領域が網点領域であるか否かを判断する工程とを備えることを特徴とする画像処理方法。

【請求項 6】 前記入力画像における網点らしさに対応する値の算出で、前記入力画像の所定領域における孤立点の中央付近ほど前記網点らしさに対応する値を高く設定することを特徴とする請求項 1 または 2 記載の画像処理装置。

【請求項 7】 入力画像から網点領域を検出するにあたり、

各々異なるウィンドウサイズによって前記入力画像の所定領域の網点らしさに対応する値を算出するステップと、

各々算出された網点らしさに対応する値に基づき前記所定領域が網点領域であるか否かを判断するステップとをコンピュータに実行させるプログラムを記録したコンピ

ュータ読み取り可能な記録媒体。

【請求項 8】 入力画像から網点領域を検出するにあたり、

各々異なるスクリーン線数によって前記入力画像の所定領域の網点らしさに対応する値を算出するステップと、各々算出された網点らしさに対応する値に基づき前記所定領域が網点領域であるか否かを判断するステップとをコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体。

10 【請求項 9】 前記入力画像における網点らしさに対応する値を算出するステップで、前記入力画像の所定領域における孤立点の中央付近ほど前記網点らしさに対応する値を高く設定することを特徴とする請求項 1 または 2 記載のコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、入力画像の所定領域が網点領域であるか否かを検知する画像処理装置、画像処理方法および画像処理プログラムを記録したコンピュータ読み取り可能な記録媒体に関する。

【0002】

【従来の技術】複写機等の画像処理装置では、原稿画像を電子データとして取り込み、種々の画像処理を施して用紙等の媒体へ印刷出力している。近年では、読み取る原稿の種類に応じて最適な画像処理を施し、原稿画像を忠実に再現できる出力を得る工夫が成されている。

【0003】例えば、原稿画像の読み取りデータから文字領域と絵柄（中間調）領域とを切り分けて、各々の領域に適した画像処理を施して出力し、文字領域はより鮮明に、また絵柄領域はより豊かな階調表現で印刷できるようにしている。

【0004】図 38 は、従来の画像処理装置を説明するブロック図である。このブロック図では、入力画像から網点領域を検出して、文字領域と絵柄領域とを分離判定する部分を示している。

【0005】つまり、入力画像に対して網点検出部 1' で網点領域の検出を行い、エッジ検出器 2 で文字などのエッジ領域を検出する。そして、これらの検出結果を AND 処理部 3 で AND 処理し、エッジ領域の場合には文字領域として「1」、網点領域の場合には絵柄（中間調）領域として「0」を Tag 生成部 4 から出力する。

【0006】ここで、網点検出部 1' での網点検出処理について説明する。網点検出部 1' では、先ず、図 39 に示すような 5×5 サイズのウィンドウを用意し、注目が素と上下左右斜めから成る八方向の画素との差の絶対値を求め、閾値処理（閾値 Th）を行い、以下の式の Ea? および Eb? の値を決定する。

【0007】

50 if (|X-X₀₀|>=Th) then Eal=1 else Eal=0

```

if (|X-X04|>=Th) then Ea2=1 else Ea2=0
if (|X-X44|>=Th) then Ea3=1 else Ea3=0
if (|X-X40|>=Th) then Ea4=1 else Ea4=0
if (|X-X02|>=Th) then Eb1=1 else Eb1=0
if (|X-X24|>=Th) then Eb2=1 else Eb2=0
if (|X-X42|>=Th) then Eb3=1 else Eb3=0
if (|X-X20|>=Th) then Eb4=1 else Eb4=0

```

【0008】次に、上記式のEa1~Ea4、Eb1~Eb4をそれぞれ加算して、Aa、Bbを求める。

Aa=Ea1+Ea2+Ea3+Ea4

Bb=Eb1+Eb2+Eb3+Eb4

【0009】そして、次の条件分岐処理を行い、ON (Haltone) の場合は網点、そうでない場合は非網点とする。

```

if (Aa==4) && (Bb==0) then Haltone else other
if (Aa==0) && (Bb==4) then Haltone else other
if (Aa==4) && (Bb==4) then Haltone else other

```

【0010】

【発明が解決しようとする課題】しかしながら、このような画像処理装置には次のような問題がある。すなわち、入力画像の網点としてスクリーン角度、スクリーン線数の異なるものが与えられたり、また入力画像の解像度が異なると画像処理上の網点におけるスクリーン角度、スクリーン線数が異なり、的確な網点検出を行えないという問題が生じる。

【0011】

【課題を解決するための手段】本発明はこのような課題を解決するために成されたものである。すなわち、本発明は、入力画像から網点領域を検出する画像処理装置であり、各々異なるウィンドウサイズもしくは各々異なるスクリーン線数によって入力画像の所定領域の網点らしさに対応する値を算出する複数の網点検知手段と、複数の網点検知手段で各々算出される網点らしさに対応する値に基づき所定領域が網点領域であるか否かを判断する判断手段とを備えている。

【0012】また、本発明は、入力画像から網点領域を検出する画像処理方法であり、各々異なるウィンドウサイズもしくは各々異なるスクリーン線数によって入力画像の所定領域の網点らしさに対応する値を算出する工程と、各々算出された網点らしさに対応する値に基づき所定領域が網点領域であるか否かを判断する工程とを備えている。

【0013】また、本発明は、入力画像から網点領域を検出するにあたり、各々異なるウィンドウサイズもしくは各々異なるスクリーン線数によって入力画像の所定領域の網点らしさに対応する値を算出するステップと、各々算出された網点らしさに対応する値に基づき所定領域が網点領域であるか否かを判断するステップとをコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体でもある。

【0014】このような本発明では、各々異なるウィンドウもしくは各々異なるスクリーン線数によって入力画像の所定領域の網点らしさに対応する値を算出し、これら算出された網点らしさに対応する値に基づき所定領域が網点領域であるか否かを判断しているため、網点のスクリーン角度やスクリーン線数が異なる場合でも的確に網点領域であることを検出できるようになる。

【0015】

【発明の実施の形態】以下、本発明の実施の形態を図に基づいて説明する。図1は、本実施形態に係る画像処理装置を説明するブロック図である。すなわち、本実施形態における画像処理装置は、入力画像（ここでは、600dpi、L*a*b*信号のうちL*信号）から網点領域を検出するものであり、主として網点抽出部1、エッジ検出器2、AND処理部3、Tag生成部4を備えている。

【0016】このうち、本実施形態の画像処理装置は網点抽出部1に特徴がある。この網点抽出部1は、浮動二値化処理部11、網点検知部(9×9)12、25×25加算部13、閾値処理部14、網点検知部(5×5)22、17×17加算部23、閾値処理部24、合成部15、膨張処理部16から構成される。

【0017】つまり、本実施形態の画像処理装置における網点抽出部1は、各々異なるウィンドウサイズによって入力画像の所定領域における網点らしさに対応する値を算出し、この各々算出された値に基づき所定領域が網点領域であるか否かを判断している。

【0018】図2は、網点抽出部の構成を説明するブロック図である。9×9のウィンドウサイズによって網点らしさに対応する値を算出する網点検知部12は、ドット配置検出部121、孤立点検出部122、排他論理の反転部123、線検知部124および条件演算&加算部125から構成され、5×5のウィンドウサイズによって網点らしさに対応する値を算出する網点検知部22は、ドット配置検出部221、孤立点検出部222、線検知部224および条件演算&加算部225から構成される。

【0019】なお、ここでは、網点検知部12の構成と網点検知部22の構成とが異なっているが、同じでもよく、また、これら以上の網点検知部を備えていてもよい。

【0020】網点検知部12では、網点らしさを判定するために、ドット配置検出部121からscoreが出力され、孤立点検出部122からscoreが出力され、排他論理の反転部123からnscoreが出力され、線検知部124からValue2が出力され、各々条件演算&加算部125で処理される。この条件演算&加算部125で演算される値は9×9サイズのウィンドウにおける注目画素の網点らしさに対応する値となり、25×25加算部13へ送られて、ここで25×25画素サイズ分の網点らしさ

に対応する値としてまとめられる。そして、 25×25 画素サイズ分の網点らしさに対応する値が閾値処理部 14 で閾値処理され、閾値領域の場合は ON、非閾値領域の場合は OFF として合成部 15 へ送られる。

【0021】一方、網点検知部 22 では、網点らしさを判定するために、ドット配置検出部 221 から score が出力され、孤立点検出部 222 から score が出力され、線検知部 224 から Value2 が出力され、各々条件演算 & 加算部 225 で処理される。この条件演算 & 加算部 225 で演算される値は 5×5 サイズのウィンドウにおける注目画素の網点らしさに対応する値となり、 17×17 加算部 23 へ送られて、ここで 17×17 画素サイズ分の網点らしさに対応する値としてまとめられる。そして、 17×17 画素サイズ分の網点らしさに対応する値が閾値処理部 24 で閾値処理され、閾値領域の場合は ON、非閾値領域の場合は OFF として合成部 15 へ送られる。

【0022】合成部 15 では、具体的には or 処理がなされ、いずれか一方の閾値処理部 14、24 から網点領域である ON 信号が送られてきた場合には、網点領域であると判定して膨張処理部 16 へその結果を出力する。その後は、網点領域であるか、非網点領域であるかに応じて適した画像信号処理を施していく。

【0023】このように、各々異なるウィンドウサイズによって入力画像の所定領域における網点らしさに対応する値を算出し、これらの値に基づき網点領域であるか否かを判断することで、網点のスクリーン角度やスクリーン線数が異なる場合でも的確に網点領域であることを検出できるようになる。

【0024】次に、本実施形態での網点らしさに対応する値の算出について説明する。まず、算出処理の概要について説明する。

【0025】(1) 始めに、図 3 に示すような 9×9 サイズのウィンドウ内において注目画素 ($X=X_{44}$) と図 4 ~ 図 15 のドットで示す各画素との排他論理和 (記号 (+) で表す) を求め、ON ならば 1、OFF ならば 0 とする。この処理を、 $3 \times 3 \sim 9 \times 9$ 領域内のそれぞれについて行う。

【0026】(2) 次いで、(1) で得られる ON の数を合計し、各領域 ($3 \times 3 \sim 9 \times 9$) の Sum 値を以下の式によって算出する。

【0027】 3×3 領域

$Sum0 = X(+)X_{34} + X(+)X_{45} + X(+)X_{54} + X(+)X_{43}$ … 図 4 参照

【0028】 5×5 領域

$if (Sum0 == 4 \ \&\& \ NS == 4 \ \&\& \ Value2 == 0)$

$out0 = 2;$

$else \ if (Sum0 == 3 \ \&\& \ NS == 4 \ \&\& \ Value2 == 0 \ \&\& \ Value2_ty == 0)$

$out0 = 1;$

$else$

$out0 = 0;$

* $Sum1 = X(+)X_{24} + X(+)X_{46} + X(+)X_{64} + X(+)X_{42}$ … 図 5 参照

【0029】 7×7 領域

$Sum2 = X(+)X_{14} + X(+)X_{47} + X(+)X_{74} + X(+)X_{41}$ … 図 6 参照

$Sum21 = X(+)X_{04} + X(+)X_{37} + X(+)X_{73} + X(+)X_{30}$ … 図 7 参照

$Sum22 = X(+)X_{24} + X(+)X_{57} + X(+)X_{84} + X(+)X_{51}$ … 図 8 参照

$Sum23 = X(+)X_{15} + X(+)X_{48} + X(+)X_{75} + X(+)X_{42}$ … 図 9 参照

$Sum24 = X(+)X_{13} + X(+)X_{46} + X(+)X_{73} + X(+)X_{40}$ … 図 10 参照

【0030】 9×9 領域

$Sum3 = X(+)X_{04} + X(+)X_{48} + X(+)X_{84} + X(+)X_{40}$ … 図 11 参照

10 $Sum31 = X(+)X_{05} + X(+)X_{85} + X(+)X_{41}$ … 図 12 参照

$Sum32 = X(+)X_{03} + X(+)X_{47} + X(+)X_{83}$ … 図 13 参照

$Sum33 = X(+)X_{38} + X(+)X_{74} + X(+)X_{30}$ … 図 14 参照

$Sum34 = X(+)X_{14} + X(+)X_{58} + X(+)X_{50}$ … 図 15 参照

【0031】また、排他論理の反転部 123 から出力される nscore (NS) は以下の式で表される。

$NS = |X(+)X_{00}| + |X(+)X_{08}| + |X(+)X_{88}| + |X(+)X_{80}|$ この式で | は反転を意味する。

【0032】(3) 次に、合計値 (Sum??) と後述する線検知部 124 からの出力 (Value2、線分の場合 1、非線分の場合 0) を基に下記条件分岐処理を行い、注目画素の網点らしさに対応するスコア値 out?? を算出する。また、パターン増加による文字誤検知低減のため、NS もこの処理に追加する。

【0033】～条件分岐処理～

各ドット配置検知に共通なスコア付け (網点検知) 処理内容

この処理により、注目画素周辺のドット配置が周期性の高いものであるほど、高いスコア値を注目画素に与える。

1) 上下左右 4 方向全ての排他論理和が ON である場合 … 2 をスコア付け (最高値)

2) 上下左右 4 方向のうち 3 方向の排他論理和が ON であり、線検知結果が 0 かつ NS が 4 方向 ON である場合 … 1 をスコア付け

3) 上記 1)、2) 以外 … 0 をスコア付け

【0034】(4) 次に、(3) で得られる各領域の out?? 値を加算し、この値を注目画素のドット配置パターンスコア (OutDotPat) 値とする。

$OutDotPat = out0 + out1 + out2 + (out21 + out22 + out23 + out24) + (out3 + out31 + out32 + out33 + out34)$

【0035】ここで、各領域における out 値の算出について具体的に説明する。

【0036】 3×3 領域の場合は以下の式によって out0 を算出する。

*

【0037】5×5領域の場合は以下の式によってout1* *を算出する。

```
if (Sum1==4 && NS==4 && Value2==0)
    out1=2;
else if (Sum1==3 && NS==4 && Value2==0 && Value2_ty==0)
    out1=1;
else
    out1=0;
```

【0038】7×7領域の場合は以下の式によってout 2, out21, out22, out23, out24を算出する。なお、7×7 ※置パターンに自由度を持たせるため、基本となるパターンに対して上下左右4方向に各々シフトしたパターンを領域の場合には、検知参照画素の増加に伴い、ドット配※10 追加している。

```
if (Sum2==4)
    out2=2;
else if (Sum2==3 && NS==4 && Value2==0)
    out2=1;
else
    out2=0;
if (Sum21==4)
    out21=2;
else if (Sum21==3 && NS==4 && Value2==0)
    out21=1;
else
    out21=0;
if (Sum22==4)
    out22=2;
else if (Sum22==3 && NS==4 && Value2==0)
    out22=1;
else
    out22=0;
if (Sum23==4)
    out23=2;
else if (Sum23==3 && NS==4 && Value2==0)
    out23=1;
else
    out23=0;
if (Sum24==4)
    out24=2;
else if (Sum24==3 && NS==4 && Value2==0)
    out24=1;
else
    out24=0;
```

【0039】9×9領域の場合は以下の式によってout3を算出する。なお、9×9領域の場合には、検知参照画素の増加に伴い、ドット配置パターンに自由度を持たせるため、基本となるパターンに対して上下左右4方向に★

★各々シフトしたパターンを追加しているが、7×7領域のドット配置検知と異なり、上下左右方向にシフトした配置パターンは3画素しかないので、ON数の最大値を3としている。

```
if (Sum3==4)
    out3=2;
else if ((Sum31==3 || Sum32==3 || Sum33==3 || Sum34==3) && NS==4 && Value2==0)
    out3=1;
else if (Sum3==3 && NS==4 && Value2==)
```

```
out3=1;
```

```
else
```

```
out3=0;
```

【0040】次に、孤立点検知部122の処理について具体的に説明する。上記説明したドット配置検知部121の処理では、計算上検知できない75線（45度）網点印刷を検知するために、この孤立点検知部122が用いられる。すなわち、9×9ウィンドウ内において領域最外周の黒画素数（図16～図19参照）をカウントし、その値（kori_3～kori_9）と網点ドットが3×3、4×4、5×5、6×6の各領域に収まるように設定した値とを大小比較（または、一致、不一致）処理を行い、ドットが孤立点であるか否かを判別する。

【0041】処理の概要は以下になる。

（1）各孤立点検知の前処理として、下記3×3～9×9各ウィンドウにおいて外周領域の各々のON（黒）画素数を以下の式によってカウントする。

【0042】3×3領域最外周のON（黒画素）数カウント：kori_3+3

$$\text{kori_3+3} = X_{33} + X_{34} + X_{35} + X_{45} + X_{55} + X_{54} + X_{53} + X_{43}$$

【0043】5×5領域最外周のON（黒画素）数カウント：kori_5+5

$$\text{kori_5+5} = X_{22} + X_{23} + X_{24} + X_{25} + X_{26} + X_{36} + X_{46} + X_{56} + X_{66} + X_{65} + X_{64} + X_{63} + X_{62} + X_{52} + X_{42} + X_{32}$$

【0044】7×7領域最外周のON（黒画素）数カウント：kori_7+7

$$\text{kori_7+7} = X_{11} + X_{12} + X_{13} + X_{14} + X_{15} + X_{16} + X_{17} + X_{27} + X_{37} + X_{47} + X_{57} + X_{67} + X_{77} + X_{76} + X_{75} + X_{74} + X_{73} + X_{72} + X_{71} + X_{61} + X_{51} + X_{41} + X_{31} + X_{21}$$

【0045】9×9領域最外周のON（黒画素）数カウント：kori_9+9

$$\text{kori_9+9} = X_{00} + X_{01} + X_{02} + X_{03} + X_{04} + X_{05} + X_{06} + X_{07} + X_{08} + X_{18} + X_{28} + X_{38} + X_{48} + X_{58} + X_{68} + X_{78} + X_{88} + X_{87} + X_{86} + X_{85} + X_{84} + X_{83} + X_{82} + X_{81} + X_{80} + X_{70} + X_{60} + X_{50} + X_{40} + X_{30} + X_{20} + X_{10}$$

【0046】（2）孤立点検知処理

下記の各領域の黒画素の個数、注目画素に対する9×9外枠4画素の排他論理和のON数の値に応じて各領域にドットが納まるか納まらないかの判別を行う。なお、下記（2-1）～（2-5）まで順番に検知を行い、検知されなかった画素については孤立点検知スコア値（Out75L）を0とする。

【0047】

（2-1）孤立点検知…3×3領域（Low）

（2-2）孤立点検知…3×3領域（High）

（2-3）孤立点検知…4×4領域

（2-4）孤立点検知…5×5領域

（2-5）孤立点検知…6×6領域

【0048】次に、孤立点検知処理の一例を以下に説明する。ここでは、孤立点ドット（浮動二値化結果）が4×4領域に納まる場合を例とする。網点検知部には、図

20～図24に示すような信号が入力されたとする。まず、これを各領域（3×3、5×5、7×7、9×9最外周）に分けて、各々黒画素のカウントを行う。

【0049】次いで、各領域の黒画素カウント数が以下の条件1に当てはまるか否かを判断する。

（条件1）

3×3領域最外周での黒画素数…8画素（すべて黒）

5×5領域最外周での黒画素数…7画素

7×7領域最外周での黒画素数…0画素

9×9領域最外周での黒画素数…0画素

【0050】この条件1を全て満たす場合には孤立点であると判定し、スコア付けを行う。

【0051】また、ドットが右へ1画素ずれた場合においても上記条件1によってドットが孤立点か否かを判定することが可能である。

【0052】さらにドットが右へシフトした場合、3×3領域が上記条件1に一致しないため、孤立点の一部であると判定できないことがある。そこで、上記条件1を満たさない場合でも、以下の条件2を満たすか否かを調べ、満たす場合には上記条件1を満たす場合の半分の値のスコア付けを行う。

【0053】（条件2）

3×3領域最外周での黒画素数…4画素以上

5×5領域最外周での黒画素数…5画素以下

7×7領域最外周での黒画素数…8画素以下

9×9領域最外周での黒画素数…0画素（すべて白）

【0054】また、このようなスコア付けを行うにあたり、図24に示すようなドット中央部では高いスコア値を与え、ドットエッジエリアについては低いスコア値を与えるようにする。こうすることにより、文字の一部が孤立点として検知されスコア付けされた場合においてもスコア値を低くできるため、後段の25×25加算&閾値処理部において非網点と判定できるようになる。

【0055】次に、孤立点検知の具体例を説明する。図25は、75線（45度）5%網点浮動二値化結果を示している。この場合、以下の条件aを全て満たす場合には孤立点としてOut75Lにスコア付けを行う。つまり、5%網点では、3×3サイズのウィンドウ内に各ドットが納まることから、以下の条件aを全て満たすことで孤立点を判定できる。

【0056】（条件a）

注目画素…ON

3×3領域最外周での黒画素数<=4

5×5領域最外周での黒画素数<=1

7×7領域最外周での黒画素数==0

9×9領域最外周での黒画素数==0

Out75L=14

【0057】図26は、75線（45度）80%網点浮動二値化結果を示している。この場合、以下の条件bを全て満たす場合には孤立点としてOut75Lにスコア付けを行う。つまり、80%網点では、3×3サイズのウィンドウ内に各ドットが納まることから、以下の条件aを全て満たすことで孤立点を判定できる。

【0058】（条件b）

注目画素…OFF

3×3領域最外周での黒画素数>=4

5×5領域最外周での黒画素数==16

7×7領域最外周での黒画素数==24

9×9領域最外周での黒画素数==32

Out75L=40

【0059】なお、これらの条件a、bに当てはまらない場合は次の条件によって孤立点検知を行う。

【0060】図27は、75線（45度）10%網点浮動二値化結果を示している。この場合、以下の条件cを全て満たす場合には孤立点としてOut75Lにスコア付けを行う。つまり、10%網点について3×3サイズのウィンドウ内に各ドットが納まることから、以下の条件cを全て満たすことで孤立点を判定できる。なお、スコア値Out75Lについての閾値は、5%網点に比べて黒画素の占有率が上昇しているため、これを考慮して設定する。

【0061】（条件c）

注目画素…ON

3×3領域最外周での黒画素数>=3

5×5領域最外周での黒画素数<=5

7×7領域最外周での黒画素数==0

9×9領域最外周での黒画素数==1

Out75L=7

【0062】図28は、75線（45度）70%網点浮動二値化結果を示している。この場合、以下の条件dを全て満たす場合には孤立点としてOut75Lにスコア付けを行う。つまり、70%網点については、3×4（4×3）サイズのウィンドウ内に各ドットが納まることから、以下の条件dを全て満たすことで孤立点を判定できる。

【0063】（条件d）

注目画素…OFF

3×3領域最外周での黒画素数<=5

5×5領域最外周での黒画素数>=（16-5）

7×7領域最外周での黒画素数==24

9×9領域最外周での黒画素数==（32-2）

Out75L=7

【0064】なお、これらの条件c、dに当てはまらない場合は次の条件によって孤立点検知を行う。

【0065】図29は、75線（45度）20%網点浮動二値化結果を示している。この場合、以下の条件eを全て満たす場合、もしくは条件e'を全て満たす場合には孤立点としてOut75Lにスコア付けを行う。つまり、2

0%網点については、4×4サイズのウィンドウ内に各ドットが納まることから、以下の条件eを全て満たすか、もしくは条件e'を全て満たすことで孤立点を判定できる。なお、ドットサイズが大きくなり、条件eだけではドット中央しか検出できないことから、条件e'を追加している。ただし、文字誤検出を防ぐため、スコアを低く（中央部の半分）にしている。

【0066】（条件e）

注目画素…ON

10 3×3領域最外周での黒画素数==8

5×5領域最外周での黒画素数<=7

7×7領域最外周での黒画素数==0

9×9領域最外周での黒画素数<=2

Out75L=7

【0067】（条件e'）

注目画素…ON

3×3領域最外周での黒画素数>=5

5×5領域最外周での黒画素数<=9

7×7領域最外周での黒画素数==（24-8）

20 9×9領域最外周での黒画素数<=4

9×9領域最外周4画素での排他論理和のON画素数==4

Out75L=3

【0068】図30は、75線（45度）60%網点浮動二値化結果を示している。この場合、以下の条件fを全て満たす場合、もしくは条件f'を全て満たす場合には孤立点としてOut75Lにスコア付けを行う。つまり、60%網点については、4×4サイズのウィンドウ内に各ドットが納まることから、以下の条件fを全て満たす場合、もしくは条件f'を全て満たす場合に孤立点を判定できる。なお、ドットサイズが大きくなり、条件fだけではドット中央しか検出できないことから、条件f'を追加している。ただし、文字誤検出を防ぐため、スコアを低く（中央部の半分）にしている。

（条件f）

注目画素…OFF

3×3領域最外周での黒画素数==0

5×5領域最外周での黒画素数>=（16-7）

7×7領域最外周での黒画素数<=24

40 9×9領域最外周での黒画素数==（32-2）

Out75L=7

【0069】（条件f'）

注目画素…OFF

3×3領域最外周での黒画素数<=3

5×5領域最外周での黒画素数>=（16-7）

7×7領域最外周での黒画素数==24

9×9領域最外周での黒画素数==（32-2）

Out75L=3

【0070】なお、これらの条件e、e'、f、f'に当てはまらない場合は次の条件によって孤立点検知を行

う。

【0071】図31は、75線（45度）30%網点浮動二値化結果を示している。この場合、以下の条件gを全て満たす場合、もしくは条件g'を全て満たす場合には孤立点としてOut75Lにスコア付けを行う。つまり、30%網点については5×5サイズのウィンドウ内に各ドットが納まることから、以下の条件gを全て満たす場合、もしくは条件g'を全て満たす場合に孤立点を判定できる。なお、ドットサイズが大きくなり、条件gだけではドット中央しか検出できないことから、条件g'を追加している。ただし、文字誤検出を防ぐため、スコアを低く（中央部の半分）にしている。

【0072】（条件g）

注目画素…ON

3×3領域最外周での黒画素数==8

5×5領域最外周での黒画素数<=7

7×7領域最外周での黒画素数==0

9×9領域最外周での黒画素数<=2

Out75L=7

【0073】（条件g'）

注目画素…ON

3×3領域最外周での黒画素数>=5

5×5領域最外周での黒画素数<=9

7×7領域最外周での黒画素数== (24-8)

9×9領域最外周での黒画素数<=4

9×9領域最外周4画素での排他論理和のON画素数==4

Out75L=3

【0074】図32は、75線（45度）50%網点浮動二値化結果を示している。この場合、以下の条件hを全て満たす場合、もしくは条件h'を全て満たす場合には孤立点としてOut75Lにスコア付けを行う。つまり、50%網点については、4×5または5×4サイズのウィンドウ内に各ドットが納まることから、以下の条件hを全て満たす場合、もしくは条件h'を全て満たす場合に孤立点を判定できる。なお、ドットサイズが大きくなり、条件hだけではドット中央しか検出できないことから、条件h'を追加している。ただし、文字誤検出を防ぐため、スコアを低く（中央部の半分）にしている。

【0075】（条件h）

注目画素…OFF

3×3領域最外周での黒画素数==0

5×5領域最外周での黒画素数>= (16-7)

7×7領域最外周での黒画素数<=24

9×9領域最外周での黒画素数== (32-2)

Out75L=7

【0076】（条件h'）

注目画素…OFF

3×3領域最外周での黒画素数<=3

5×5領域最外周での黒画素数>= (16-7)

7×7領域最外周での黒画素数==24

9×9領域最外周での黒画素数== (32-2) 4

9×9領域最外周4画素での排他論理和のON画素数==4

Out75L=3

【0077】なお、これらの条件g, g', h, h'に当てはまらない場合は次の条件によって孤立点検知を行う。

【0078】図33は、75線（45度）40%網点浮動二値化結果を示している。この場合、以下の条件iを全て満たす場合、もしくは条件i'を全て満たす場合には孤立点としてOut75Lにスコア付けを行う。つまり、40%網点については6×6サイズのウィンドウ内に各ドットが納まることから、以下の条件iを全て満たす場合、もしくは条件i'を全て満たす場合に孤立点を判定できる。なお、ドットサイズが大きくなり、条件iだけではドット中央しか検出できないことから、条件i'を追加している。ただし、文字誤検出を防ぐため、スコアを低く（中央部の半分）にしている。

20 【0079】（条件i）

注目画素…ON

3×3領域最外周での黒画素数==8

5×5領域最外周での黒画素数<=7

7×7領域最外周での黒画素数==0

9×9領域最外周での黒画素数<=2

Out75L=7

【0080】（条件i'）

注目画素…ON

3×3領域最外周での黒画素数>=5

30 5×5領域最外周での黒画素数<=9

7×7領域最外周での黒画素数== (24-8)

9×9領域最外周での黒画素数<=4

9×9領域最外周4画素での排他論理和のON画素数==4

Out75L=3

【0081】図34は、75線（45度）40%網点浮動二値化結果を示している。この場合、以下の条件jを全て満たす場合、もしくは条件j'を全て満たす場合には孤立点としてOut75Lにスコア付けを行う。つまり、40%網点については、6×6サイズのウィンドウ内に各ドットが納まることから、以下の条件jを全て満たす場合、もしくは条件j'を全て満たす場合に孤立点を判定できる。なお、ドットサイズが大きくなり、条件jだけではドット中央しか検出できないことから、条件j'を追加している。ただし、文字誤検出を防ぐため、スコアを低く（中央部の半分）にしている。

【0082】（条件j）

注目画素…OFF

3×3領域最外周での黒画素数==0

50 5×5領域最外周での黒画素数>= (16-7)

7×7領域最外周での黒画素数<=24
 9×9領域最外周での黒画素数==(32-2)
 Out75L=7
 【0083】(条件j')
 注目画素…OFF
 3×3領域最外周での黒画素数<=3
 5×5領域最外周での黒画素数>=(16-7)
 7×7領域最外周での黒画素数==24
 9×9領域最外周での黒画素数==(32-2)
 9×9領域最外周4画素での排他論理和のON画素数== 10
 4
 Out75L=3

【0084】なお、上記条件のいずれにも当てはまらない場合は孤立点ではないとして、Out75L=0を設定する。

【0085】次に、線検知部124での処理について具体的に説明する。すなわち、処理手順としては以下のようになる。

【0086】(1) 9×9サイズnウィンドウ内において対角線上(図35の斜線部参照)に存在する黒画素数Line1、Line2を以下の式を用いてカウントする。

【0087】

Line1=X₀₀+X₁₁+X₂₂+X₃₃+X₄₄+X₅₅+X₆₆+X₇₇+X₈₈
 Line2=X₈₀+X₇₁+X₆₂+X₅₃+X₄₄+X₃₅+X₂₆+X₁₇+X₀₈

【0088】(2) 9×9サイズのウィンドウ内において注目画素を含む縦ライン(図36の斜線部参照)の黒画素数Line3、Line4を以下の式を用いてカウントする。

【0089】

Line3=X₄₀+X₄₁+X₄₂+X₄₃+X₄₄+X₄₅+X₄₆+X₄₇+X₄₈
 Line4=X₈₄+X₇₄+X₆₄+X₅₄+X₄₄+X₃₄+X₂₄+X₁₄+X₀₄

【0090】(3) 9画素連続して黒または白がある場合、線分の一部と判定する以下の条件分岐を用いて線分/非線分の検知を行う。

【0091】

条件分岐処理(対角線上)

if(Line1==9 || Line2==9 || Line1==0 || Line2==0)

Value2=1

else

Value2=0

【0092】

条件分岐処理(注目画素を含む縦横ライン)

if(Line3==9 || Line4==9 || Line3==0 || Line4==0)

Value2_ty=1

else

Value2_ty=0

【0093】次に、排他論理の反転部124から出力されるnscoreについて具体的に説明する。図37は、6p明朝文字「場」の一部の浮動二値化結果を示す図である。このようなデータでは、上記説明したドット配置検知部で網点として検知してしまう場合がある。

【0094】そこで、9×9サイズのウィンドウを用い、文字の斜め線部にこのウィンドウを当てはめると、四隅の部分の全てにおいて注目画素に対して右上の排他論理和がONではないことに着目し、斜め線部の誤検知を低減させるべく、nscore(NS)を算出する。

【0095】このように、図2に示すドット配置検知部121、孤立点検知部122、排他論理の反転部123および線検知部124によって各々出力されるスコア値を用い、注目画素に対する網点らしさに対応する値(最終スコア値)を以下の式によって算出する。

【0096】最終スコア値=OutDotPat+Out75L

ここで、OutDotPat=out0+out1+out2+(out21+out22+out23+out24)+(out3+out31+out32+out33+out34)である。

【0097】これにより、周期性および孤立性の高い画素ほど高い最終スコア値となる。また、後段の25×25加算部13では、最終スコア値を25×25サイズのウィンドウで加算し、閾値処理部14で閾値処理して網点領域、非網点領域の分離を行うことになる。

【0098】なお、上記実施形態では、主として異なるウィンドウサイズを備えた複数の網点検出部12、22で各々網点らしさに対応する値を算出し、これらに基づき網点領域か否かを判断する例を説明したが、本発明では、異なるスクリーン線数によって網点らしさに対応する値を各々算出する複数の網点検出部を備えるようにしてもよい。また、上記説明した各部の処理は、画像処理プログラムとしてコンピュータ読み取り可能な記録媒体に格納されていてもよい。

【0099】

【発明の効果】以上説明したように、本発明によれば次のような効果がある。すなわち、各々異なるウィンドウもしくは各々異なるスクリーン線数によって入力画像の所定領域の網点らしさに対応する値を算出し、これら算出された網点らしさに対応する値に基づき所定領域が網点領域であるか否かを判断しているため、網点のスクリーン角度やスクリーン線数が異なる場合でも的確に網点領域であることを検出できるようになる。これにより、後段の処理で、網点領域に対する最適な処理を施すことが可能となる。

【図面の簡単な説明】

40 【図1】 本実施形態に係る画像処理装置を説明するブロック図である。

【図2】 網点抽出部の構成を説明するブロック図である。

【図3】 9×9サイズのウィンドウを示す図である。

【図4】 排他論理和を求めるドット配置を示す図(その1)である。

【図5】 排他論理和を求めるドット配置を示す図(その2)である。

50 【図6】 排他論理和を求めるドット配置を示す図(その3)である。

【図 7】 排他論理和を求めるドット配置を示す図（その 4）である。

【図 8】 排他論理和を求めるドット配置を示す図（その 5）である。

【図 9】 排他論理和を求めるドット配置を示す図（その 6）である。

【図 10】 排他論理和を求めるドット配置を示す図（その 7）である。

【図 11】 排他論理和を求めるドット配置を示す図（その 8）である。

【図 12】 排他論理和を求めるドット配置を示す図（その 9）である。

【図 13】 排他論理和を求めるドット配置を示す図（その 10）である。

【図 14】 排他論理和を求めるドット配置を示す図（その 11）である。

【図 15】 排他論理和を求めるドット配置を示す図（その 12）である。

【図 16】 3×3 ウィンドウの最外周を説明する図である。

【図 17】 5×5 ウィンドウの最外周を説明する図である。

【図 18】 7×7 ウィンドウの最外周を説明する図である。

【図 19】 9×9 ウィンドウの最外周を説明する図である。

【図 20】 孤立点の例を示す図（その 1）である。

【図 21】 孤立点の例を示す図（その 2）である。

【図 22】 孤立点の例を示す図（その 3）である。

【図 23】 孤立点の例を示す図（その 4）である。

【図 24】 ドット中央部に高いスコアを与える例を示す図である。

【図 25】 75 線（45 度）5% 網点浮動二値化結果を示す図である。

【図 26】 75 線（45 度）80% 網点浮動二値化結果を示す図である。

【図 27】 75 線（45 度）10% 網点浮動二値化結果を示す図である。

【図 28】 75 線（45 度）70% 網点浮動二値化結果を示す図である。

【図 29】 75 線（45 度）20% 網点浮動二値化結果を示す図である。

【図 30】 75 線（45 度）60% 網点浮動二値化結果を示す図である。

【図 31】 75 線（45 度）30% 網点浮動二値化結果を示す図である。

【図 32】 75 線（45 度）50% 網点浮動二値化結果を示す図である。

【図 33】 75 線（45 度）40% 網点浮動二値化結果を示す図である。

【図 34】 75 線（45 度）40% 網点浮動二値化結果を示す図である。

【図 35】 対角線上にある線分検知パターンを示す図である。

【図 36】 縦横ライン上にある線分検知パターンを示す図である。

【図 37】 明朝文字「場」の一部の浮動二値化結果を示す図である。

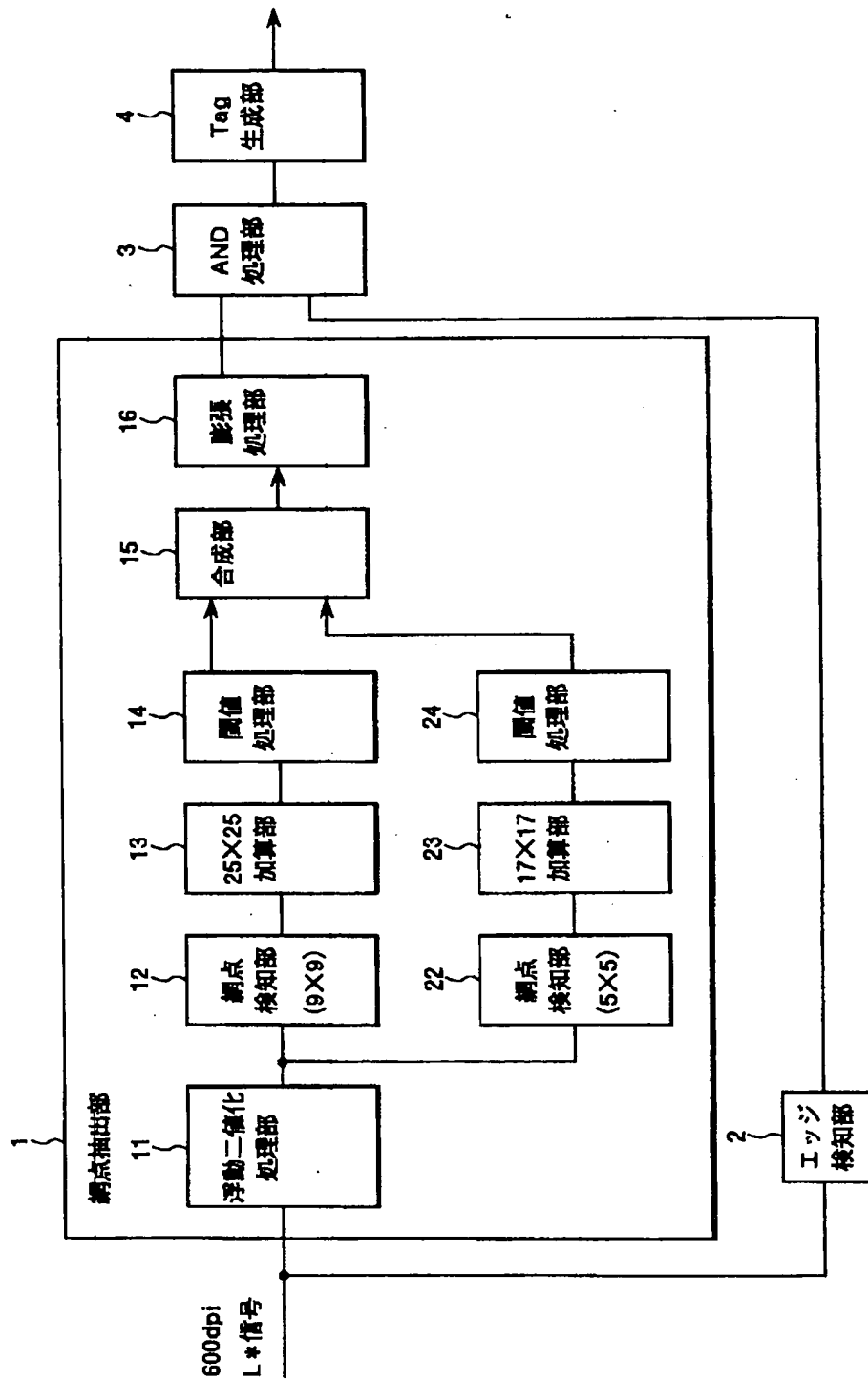
【図 38】 従来例を説明するブロック図である。

【図 39】 5×5 サイズのウィンドウを示す図である。

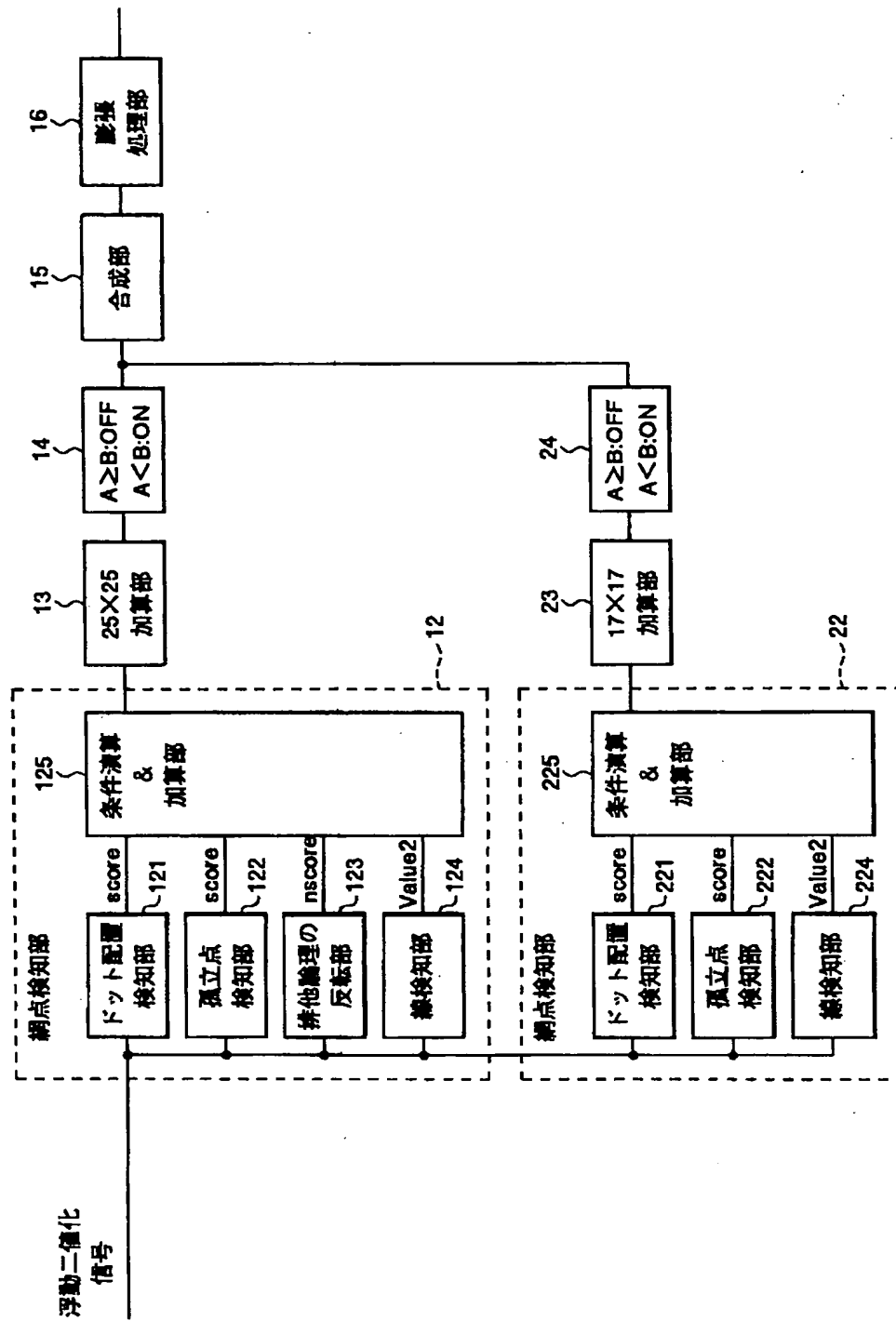
【符号の説明】

1…網点抽出部、2…エッジ検知器、3…AND 処理部、4…Tag 生成部、11…浮動二値化処理部、12…網点検知部（9×9）、13…25×25 加算部、14…閾値処理部、15…合成部、16…膨張処理部、22…網点検知部（5×5）、23…17×17 加算部、24…閾値処理部、121…ドット配置検知部、122…孤立点検知部、123…排他論理の反転部、124…線検知部、125…条件演算&加算部、221…ドット配置検知部、222…孤立点検知部、224…線検知部、225…条件演算&加算部

【図1】



【図2】



【図 3】

X00	X01	X02	X03	X04	X05	X06	X07	X08
X10	X11	X12	X13	X14	X15	X16	X17	X18
X20	X21	X22	X23	X24	X25	X26	X27	X28
X30	X31	X32	X33	X34	X35	X36	X37	X38
X40	X41	X42	X43	X44	X45	X46	X47	X48
X50	X51	X52	X53	X54	X55	X56	X57	X58
X60	X61	X62	X63	X64	X65	X66	X67	X68
X70	X71	X72	X73	X74	X75	X76	X77	X78
X80	X81	X82	X83	X84	X85	X86	X87	X88

【図 4】

X00	X01	X02	X03	X04	X05	X06	X07	X08
X10	X11	X12	X13	X14	X15	X16	X17	X18
X20	X21	X22	X23	X24	X25	X26	X27	X28
X30	X31	X32	X33	X34	X35	X36	X37	X38
X40	X41	X42	X43	X44	X45	X46	X47	X48
X50	X51	X52	X53	X54	X55	X56	X57	X58
X60	X61	X62	X63	X64	X65	X66	X67	X68
X70	X71	X72	X73	X74	X75	X76	X77	X78
X80	X81	X82	X83	X84	X85	X86	X87	X88

【図 5】

X00	X01	X02	X03	X04	X05	X06	X07	X08
X10	X11	X12	X13	X14	X15	X16	X17	X18
X20	X21	X22	X23	X24	X25	X26	X27	X28
X30	X31	X32	X33	X34	X35	X36	X37	X38
X40	X41	X42	X43	X44	X45	X46	X47	X48
X50	X51	X52	X53	X54	X55	X56	X57	X58
X60	X61	X62	X63	X64	X65	X66	X67	X68
X70	X71	X72	X73	X74	X75	X76	X77	X78
X80	X81	X82	X83	X84	X85	X86	X87	X88

【図 6】

X00	X01	X02	X03	X04	X05	X06	X07	X08
X10	X11	X12	X13	X14	X15	X16	X17	X18
X20	X21	X22	X23	X24	X25	X26	X27	X28
X30	X31	X32	X33	X34	X35	X36	X37	X38
X40	X41	X42	X43	X44	X45	X46	X47	X48
X50	X51	X52	X53	X54	X55	X56	X57	X58
X60	X61	X62	X63	X64	X65	X66	X67	X68
X70	X71	X72	X73	X74	X75	X76	X77	X78
X80	X81	X82	X83	X84	X85	X86	X87	X88

【図 7】

X00	X01	X02	X03	X04	X05	X06	X07	X08
X10	X11	X12	X13	X14	X15	X16	X17	X18
X20	X21	X22	X23	X24	X25	X26	X27	X28
X30	X31	X32	X33	X34	X35	X36	X37	X38
X40	X41	X42	X43	X44	X45	X46	X47	X48
X50	X51	X52	X53	X54	X55	X56	X57	X58
X60	X61	X62	X63	X64	X65	X66	X67	X68
X70	X71	X72	X73	X74	X75	X76	X77	X78
X80	X81	X82	X83	X84	X85	X86	X87	X88

【図 8】

X00	X01	X02	X03	X04	X05	X06	X07	X08
X10	X11	X12	X13	X14	X15	X16	X17	X18
X20	X21	X22	X23	X24	X25	X26	X27	X28
X30	X31	X32	X33	X34	X35	X36	X37	X38
X40	X41	X42	X43	X44	X45	X46	X47	X48
X50	X51	X52	X53	X54	X55	X56	X57	X58
X60	X61	X62	X63	X64	X65	X66	X67	X68
X70	X71	X72	X73	X74	X75	X76	X77	X78
X80	X81	X82	X83	X84	X85	X86	X87	X88

【図 9】

X00	X01	X02	X03	X04	X05	X06	X07	X08
X10	X11	X12	X13	X14	X15	X16	X17	X18
X20	X21	X22	X23	X24	X25	X26	X27	X28
X30	X31	X32	X33	X34	X35	X36	X37	X38
X40	X41	X42	X43	X44	X45	X46	X47	X48
X50	X51	X52	X53	X54	X55	X56	X57	X58
X60	X61	X62	X63	X64	X65	X66	X67	X68
X70	X71	X72	X73	X74	X75	X76	X77	X78
X80	X81	X82	X83	X84	X85	X86	X87	X88

【図 10】

X00	X01	X02	X03	X04	X05	X06	X07	X08
X10	X11	X12	X13	X14	X15	X16	X17	X18
X20	X21	X22	X23	X24	X25	X26	X27	X28
X30	X31	X32	X33	X34	X35	X36	X37	X38
X40	X41	X42	X43	X44	X45	X46	X47	X48
X50	X51	X52	X53	X54	X55	X56	X57	X58
X60	X61	X62	X63	X64	X65	X66	X67	X68
X70	X71	X72	X73	X74	X75	X76	X77	X78
X80	X81	X82	X83	X84	X85	X86	X87	X88

【図 11】

X00	X01	X02	X03	X04	X05	X06	X07	X08
X10	X11	X12	X13	X14	X15	X16	X17	X18
X20	X21	X22	X23	X24	X25	X26	X27	X28
X30	X31	X32	X33	X34	X35	X36	X37	X38
X40	X41	X42	X43	X44	X45	X46	X47	X48
X50	X51	X52	X53	X54	X55	X56	X57	X58
X60	X61	X62	X63	X64	X65	X66	X67	X68
X70	X71	X72	X73	X74	X75	X76	X77	X78
X80	X81	X82	X83	X84	X85	X86	X87	X88

【図 12】

X00	X01	X02	X03	X04	X05	X06	X07	X08
X10	X11	X12	X13	X14	X15	X16	X17	X18
X20	X21	X22	X23	X24	X25	X26	X27	X28
X30	X31	X32	X33	X34	X35	X36	X37	X38
X40	X41	X42	X43	X44	X45	X46	X47	X48
X50	X51	X52	X53	X54	X55	X56	X57	X58
X60	X61	X62	X63	X64	X65	X66	X67	X68
X70	X71	X72	X73	X74	X75	X76	X77	X78
X80	X81	X82	X83	X84	X85	X86	X87	X88

【図 13】

X00	X01	X02	X03	X04	X05	X06	X07	X08
X10	X11	X12	X13	X14	X15	X16	X17	X18
X20	X21	X22	X23	X24	X25	X26	X27	X28
X30	X31	X32	X33	X34	X35	X36	X37	X38
X40	X41	X42	X43	X44	X45	X46	X47	X48
X50	X51	X52	X53	X54	X55	X56	X57	X58
X60	X61	X62	X63	X64	X65	X66	X67	X68
X70	X71	X72	X73	X74	X75	X76	X77	X78
X80	X81	X82	X83	X84	X85	X86	X87	X88

【図 14】

X00	X01	X02	X03	X04	X05	X06	X07	X08
X10	X11	X12	X13	X14	X15	X16	X17	X18
X20	X21	X22	X23	X24	X25	X26	X27	X28
X30	X31	X32	X33	X34	X35	X36	X37	X38
X40	X41	X42	X43	X44	X45	X46	X47	X48
X50	X51	X52	X53	X54	X55	X56	X57	X58
X60	X61	X62	X63	X64	X65	X66	X67	X68
X70	X71	X72	X73	X74	X75	X76	X77	X78
X80	X81	X82	X83	X84	X85	X86	X87	X88

【図 15】

X ₀₀	X ₀₁	X ₀₂	X ₀₃	X ₀₄	X ₀₅	X ₀₆	X ₀₇	X ₀₈
X ₁₀	X ₁₁	X ₁₂	X ₁₃	X ₁₄	X ₁₅	X ₁₆	X ₁₇	X ₁₈
X ₂₀	X ₂₁	X ₂₂	X ₂₃	X ₂₄	X ₂₅	X ₂₆	X ₂₇	X ₂₈
X ₃₀	X ₃₁	X ₃₂	X ₃₃	X ₃₄	X ₃₅	X ₃₆	X ₃₇	X ₃₈
X ₄₀	X ₄₁	X ₄₂	X ₄₃	X ₄₄	X ₄₅	X ₄₆	X ₄₇	X ₄₈
X ₅₀	X ₅₁	X ₅₂	X ₅₃	X ₅₄	X ₅₅	X ₅₆	X ₅₇	X ₅₈
X ₆₀	X ₆₁	X ₆₂	X ₆₃	X ₆₄	X ₆₅	X ₆₆	X ₆₇	X ₆₈
X ₇₀	X ₇₁	X ₇₂	X ₇₃	X ₇₄	X ₇₅	X ₇₆	X ₇₇	X ₇₈
X ₈₀	X ₈₁	X ₈₂	X ₈₃	X ₈₄	X ₈₅	X ₈₆	X ₈₇	X ₈₈

【図 16】

【図 17】

【図 18】

【図 19】

X ₀₀	X ₀₁	X ₀₂	X ₀₃	X ₀₄	X ₀₅	X ₀₆	X ₀₇	X ₀₈
X ₁₀								X ₁₈
X ₂₀								X ₂₈
X ₃₀								X ₃₈
X ₄₀								X ₄₈
X ₅₀								X ₅₈
X ₆₀								X ₆₈
X ₇₀								X ₇₈
X ₈₀	X ₈₁	X ₈₂	X ₈₃	X ₈₄	X ₈₅	X ₈₆	X ₈₇	X ₈₈

【図 20】

X ₀₀	X ₀₁	X ₀₂	X ₀₃	X ₀₄	X ₀₅	X ₀₆	X ₀₇	X ₀₈
X ₁₀	X ₁₁	X ₁₂	X ₁₃	X ₁₄	X ₁₅	X ₁₆	X ₁₇	X ₁₈
X ₂₀	X ₂₁	X ₂₂	X ₂₃	X ₂₄	X ₂₅	X ₂₆	X ₂₇	X ₂₈
X ₃₀	X ₃₁	X ₃₂	X ₃₃	X ₃₄	X ₃₅	X ₃₆	X ₃₇	X ₃₈
X ₄₀	X ₄₁	X ₄₂	X ₄₃	X ₄₄	X ₄₅	X ₄₆	X ₄₇	X ₄₈
X ₅₀	X ₅₁	X ₅₂	X ₅₃	X ₅₄	X ₅₅	X ₅₆	X ₅₇	X ₅₈
X ₆₀	X ₆₁	X ₆₂	X ₆₃	X ₆₄	X ₆₅	X ₆₆	X ₆₇	X ₆₈
X ₇₀	X ₇₁	X ₇₂	X ₇₃	X ₇₄	X ₇₅	X ₇₆	X ₇₇	X ₇₈
X ₈₀	X ₈₁	X ₈₂	X ₈₃	X ₈₄	X ₈₅	X ₈₆	X ₈₇	X ₈₈

【图 3 9】

X ₀₀	X ₀₁	X ₀₂	X ₀₃	X ₀₄
X ₁₀	X ₁₁	X ₁₂	X ₁₃	X ₁₄
X ₂₀	X ₂₁	X ₂₂	X ₂₃	X ₂₄
X ₃₀	X ₃₁	X ₃₂	X ₃₃	X ₃₄
X ₄₀	X ₄₁	X ₄₂	X ₄₃	X ₄₄

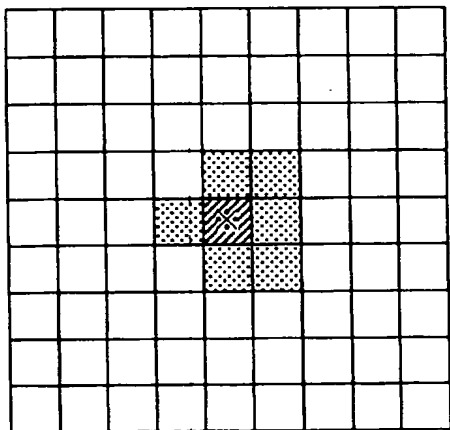
【図 23】

A 10x10 grid with a 4x4 subgrid in the center. The subgrid contains patterns of dots and diagonal lines.

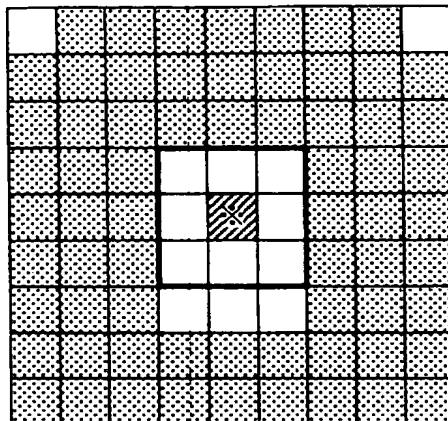
【図 25】

A 10x10 grid of squares. A 3x3 square is centered within the grid. The center square of this 3x3 square is shaded with diagonal lines.

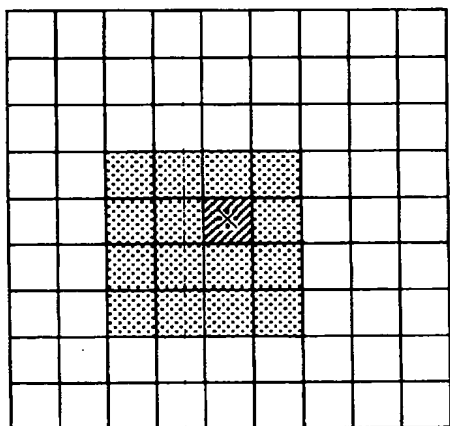
【図 27】



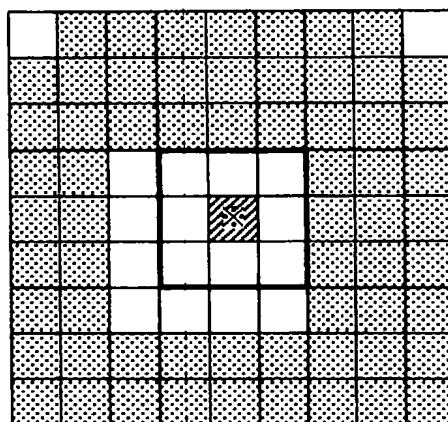
【図 28】



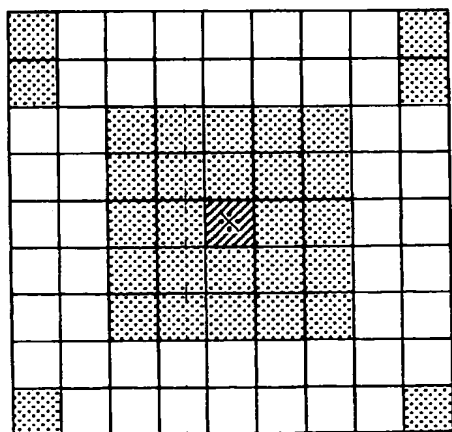
【図 29】



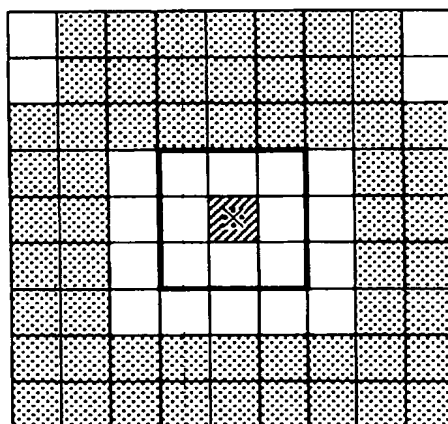
【図 30】



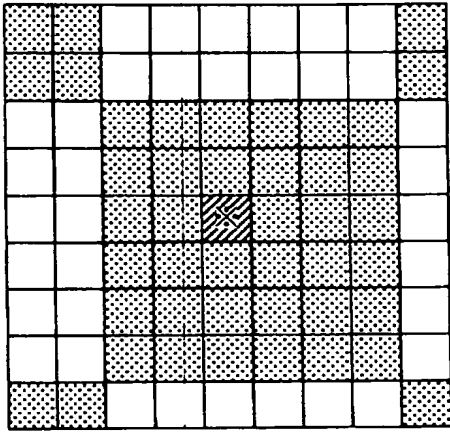
【図 31】



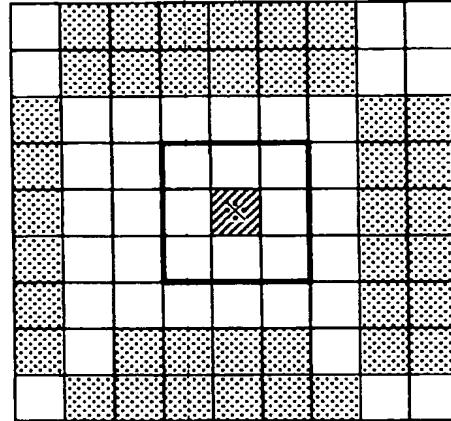
【図 32】



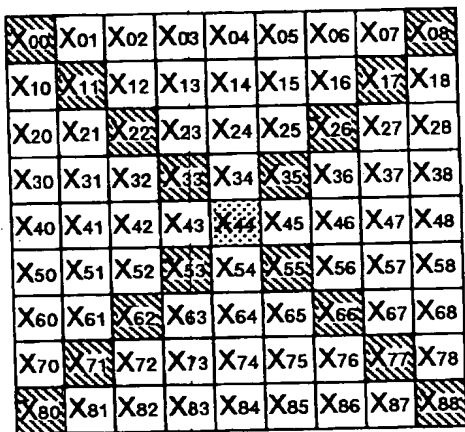
【図 3 3】



【図 3 4】



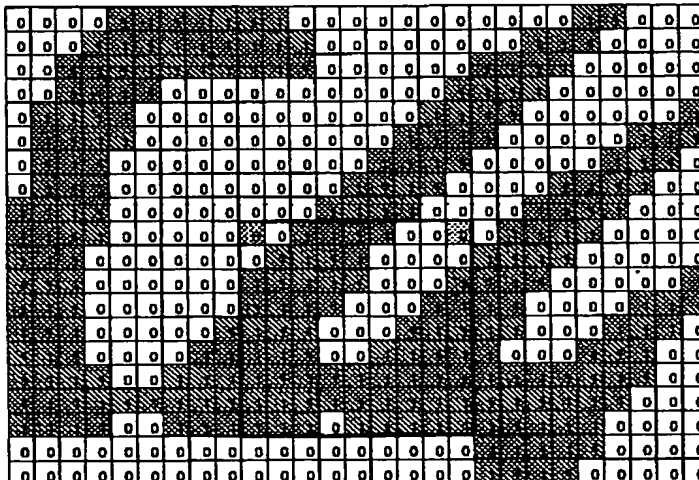
【図 3 5】



【図 3 6】

X00	X01	X02	X03	X04	X05	X06	X07	X08
X10	X11	X12	X13	X14	X15	X16	X17	X18
X20	X21	X22	X23	X24	X25	X26	X27	X28
X30	X31	X32	X33	X34	X35	X36	X37	X38
X40	X41	X42	X43	X44	X45	X46	X47	X48
X50	X51	X52	X53	X54	X55	X56	X57	X58
X60	X61	X62	X63	X64	X65	X66	X67	X68
X70	X71	X72	X73	X74	X75	X76	X77	X78
X80	X81	X82	X83	X84	X85	X86	X87	X88

【図 3 7】



【図38】

